

IN THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application.

1. (Canceled)
2. (Previously Presented) The API of claim 1 ~~37~~, wherein:
the scriptable plug-in API has a plurality of first interfaces;
the non-scriptable plug-in API has a plurality of second interfaces; and
each of said bridges connects a respective one of said first interfaces to a
respective one of said second interfaces.
3. (Currently Amended) ~~An API comprising:~~ A computer-readable memory device
encoded with a data structure providing an interface to an application program that is
encoded in the computer-readable memory device and that is run by a processor in a data
processing system, the data structure comprising:
a cross platform language API;
a scriptable language API;
a first interface operatively configured to connect said cross platform language
API and said scriptable language API;
a non-scriptable plug-in API; and

a second interface operatively configured to connect said non-scriptable plug-in API and said cross platform language API such that said scriptable language API is able to access said non-scriptable plug-in API.

4. (Cancelled).

5. (Currently Amended) The API computer-readable memory device of claim 3 further comprising:

a non-scriptable plug-in object implemented in said non-scriptable plug-in API;

a cross platform language object implemented in said cross platform language API to correspond to said non-scriptable plug-in object; and

a scripting language object implemented in said scripting language API;

wherein said second interface operatively connects said non-scriptable plug-in object and said cross platform language object, and

said first interface operatively connects said cross platform language object and said scripting language object, such that said cross platform language object operates as a proxy for said non-scriptable plug-in object.

6. (Currently Amended) The API computer-readable memory device of claim 3, wherein said first interface is an XPIDL interface.

7. (Currently Amended) The API computer-readable memory device of claim 3, wherein said second interface is an XPConnect interface.

8. (Currently Amended) The ~~API~~ computer-readable memory device of claim 7, wherein said XPConnect interface uses a typelib file associated with the cross platform language object.

9. (Currently Amended) The ~~API~~ computer-readable memory device of claim 3 wherein said scripting language object is one of a Javascript object, a Perl object, and a Python object.

10. (Currently Amended) The ~~API~~ computer-readable memory device of claim 3, wherein said cross platform language is XPCOM.

11. (Currently Amended) ~~A scriptable plug-in API comprises:~~ A computer-readable memory device encoded with a data structure providing an interface to an application program that is encoded in the computer-readable memory device and that is run by a processor in a data processing system, the date structure comprising:
a scriptable plug-in; and
a proxy support interface, wherein said scriptable plug-in is able to perform inter-thread calls through said proxy support interface.

12. (Currently Amended) The ~~scriptable plug-in API~~ computer-readable memory device of claim 11, wherein said proxy support interface is a nsISupports Proxy.

13. (Previously Presented) A method in a data processing system having a program for implementing an API, the method comprising the steps of:
obtaining a non-scriptable plug-in API;

obtaining a scriptable plug-in API; and
implementing a plurality of bridges to operatively connect said scriptable and said non-scriptable plug-in APIs such that a scriptable plug-in program is able to access the non-scriptable plug-in API in response to implementing the scriptable plug-in API.

14. (Previously Presented) The method of claim 13, wherein:
said scriptable plug-in API has a plurality of first interfaces;
said non-scriptable plug-in API has a plurality of second interfaces; and
each of said bridges connects a respective one of said first interfaces to a respective one of said second interfaces.

15. (Previously Presented) A method in a data processing system having a program for implementing an API, the method comprising the steps of:
obtaining a cross platform language API;
obtaining a scriptable language API; and
implementing a first interface to operatively connect said cross platform language API and said scriptable language API;
obtaining a non-scriptable plug-in API; and
implementing a second interface to operatively connect said non-scriptable plug-in API and said cross platform language API such that said scriptable language API is able to access said non-scriptable plug-in API.

16. (Cancelled).

17. (Previously Presented) The method of claim 15 further comprising:
defining a non-scriptable plug-in object in said non-scriptable plug-in API;
defining a cross platform language object in said cross platform language API to
correspond to said non-scriptable plug-in object; and
defining a scripting language object in said scripting language API;
wherein said second interface operatively connects said non-scriptable plug-in
object and said cross platform language object, and
said first interface operatively connects said cross platform language object and
said scripting language object, such that said cross platform language object operates as a proxy
for said non-scriptable plug-in object.
18. (Previously Presented) The method of claim 15, wherein said first interface is an
XPIDL interface.
19. (Previously Presented) The method of claim 15, wherein said second interface is
an XPConnect interface.
20. (Previously Presented) The method of claim 19, wherein said XPConnect
interface uses a typelib file associated with the cross platform language object.
21. (Previously Presented) The method of claim 15, wherein said scripting language
object is one of a Javascript object, a Perl object, and a Python object.
22. (Previously Presented) The method of claim 15, wherein said cross platform
language is XPCOM.

23. (Previously Presented) A method in data processing system having a program for implementing a scriptable plug-in API, the method comprising the steps of:

implementing a scriptable plug-in; and

generating a proxy support interface wherein said scriptable plug-in is able to perform a inter-thread call through said proxy support interface.

24. (Previously Presented) The method of claim 23, wherein said proxy support interface is a nsISupports Proxy.

25. (Currently Amended) A tangible computer readable medium containing instructions causing a program in a computer system to perform a method, the method comprising:

obtaining a non-scriptable plug-in API;

obtaining a scriptable plug-in API; and

implementing a plurality of bridges to operatively connect said scriptable and said non-scriptable plug-in APIs such that a scriptable plug-in program is able to access the non-scriptable plug-in API in response to implementing the scriptable plug-in API.

26. (Previously Presented) The computer readable medium of claim 25, wherein:

said scriptable plug-in API has a plurality of first interfaces;

said non-scriptable plug-in API has a plurality of second interfaces; and

each of said bridges to connect said first interface and said second interface.

27. (Currently Amended) A tangible computer readable medium containing instructions causing a program in a computer system to perform a method, the method comprising:

- obtaining a cross platform language API;
- obtaining a scriptable language API; and
- implementing a first interface to operatively connect said cross platform language API and said scriptable language API;
- obtaining a non-scriptable plug-in API; and
- implementing a second interface to operatively connect said non-scriptable plug-in API and said cross platform language API such that said scriptable language API is able to access said non-scriptable plug-in API.

28. (Cancelled).

29. (Previously Presented) The computer readable medium of claim 27, the method further comprising:

- defining a non-scriptable plug-in object in said non-scriptable plug-in API;
- defining a cross platform language object in said cross platform language API to correspond to said non-scriptable plug-in object; and
- defining a scripting language object in said scripting language API, wherein said first interface operatively connects said cross platform language object and said scripting language object; and

said second interface operatively connects said non-scriptable plug-in object and said cross platform language object, such that said cross platform language object operates as a proxy for said non-scriptable plug-in object.

30. (Previously Presented) The computer readable medium of claim 27, wherein said first interface is an XPIDL interface.

31. (Previously Presented) The computer readable medium of claim 27, wherein said second interface is an XPConnect interface.

32. (Previously Presented) The computer readable medium of claim 31, wherein XPConnect interface uses a typelib file associated with the cross platform language object.

33. (Previously Presented) The computer readable medium of claim 27, wherein said scripting language object is one of a Javascript object, a Perl object, and a Python object.

34. (Previously Presented) The computer readable medium of claim 27, wherein said cross platform language is XPCOM.

35. (Currently Amended) A tangible computer readable medium containing instructions causing a program in a computer system to perform a method, the method comprising:

obtaining a scriptable plug-in; and

generating a proxy support interface such that said scriptable plug-in is able to perform an inter-thread call through said proxy support interface.

36. (Previously Presented) The computer readable medium of claim 35, wherein said proxy support interface is a nsISupports Proxy.

37. (Previously Presented) A computer-readable memory device encoded with a data structure providing an interface to an application program that is encoded in the computer-readable memory device and that is run by a processor in a data processing system, the data structure comprising:

a non-scriptable plug-in API;

a scriptable plug-in API; and

a plurality of bridges operatively configured to connect said scriptable and said non-scriptable plug-in APIs such that a scriptable plug-in program is able to access the non-scriptable plug-in API in response to implementing the scriptable plug-in API.